[redacted]

5/9

array broadside due to the sequential sampling, although the parallel A/D is ideal with additional cost

4. We use the general purpose Omni-directional electret condenser microphone sensor due to its low cost, A Uni-directional sensor provides more robust noise reduction performance, but comes with more expensive price

Part 2: Microphone array software

Due to the facts that the handheld game controller:

1. **May freely move around in 3D space with six degree-of-freedom during audio recording**
2. **May be used in extremely loud gaming environment or background noises, which may include: TV, HI-FI music, voice of other players, ambient noise, etc...**
3. **Prefer simple and easy manufacture process with extremely low cost**
4. **Has little DSP capability on board**
5. **Compact mounting area available on the controller surface**

In contrast, most prior art array products, such as PC microphone array, car array, conference array (see attached microphone array introduction), they typically depend on extremely powerful DSP silicon to reach the real-time signal processing, they:

1. Usually are fixed on mounting frame (e.g. PC monitor, car interior, helmet)
2. Provide freezed beam forming (the listening direction is locked), so the talking person doesn't have much degree of freedom to move around
3. Has either limited noise cancellation capability (e.g. moderate office environment) or fixed type of noises (e.g. car engine, mechanical noise)
4. Are generally designed for voice communication, the fact that human ear has great capability to tolerance voice deformation makes the array cares very little to voice signal distortion that is crucial for computer to understand voice, such as speech recognition, speaker identification
5. Typically need relatively larger mounting area and have large amount of sensors to achieve competitive noise reduction performance

Most of those cons derives from the basic software architecture: it is traditionally build-in firmware running on expensive DSP chip with very little memory and limited computation power in order to shoot for the multi-channel real-time signal processing

However, things are different on game console, its real-time response, memory bandwidth and computational power makes it possible to be used as general purpose processor to serve even the most sophisticated real-time massive signal processing applications, hence to replace the use of the special-purpose DSP silicon, the pros [redacted]

[redacted]

1. Since the software runs on console, the cost is possibly lowered down to cents level, instead of hundreds bulks on average on market. This conquers the fact 3: the cost issue
2. Without the DSP hardware limitation (the fact 4), the complexity of noise reduction software can be extend to an state-of-art edge that was only available in academia research in the past, it is in fact, in a sense that we have effectively transformed the traditional software design problem from figuring out the engineering tedious (such as, speed, fast response, code size) to an unlimited functional sky.

Given the great computational powerhouse we may possesses on future game console platform, the software designed takes full advantageous of today's cutting-edge noise-reduction algorithms, it integrates:

1. Acoustic Echo Cancellation (AEC)
-- To reduce noise generated from the console sound output

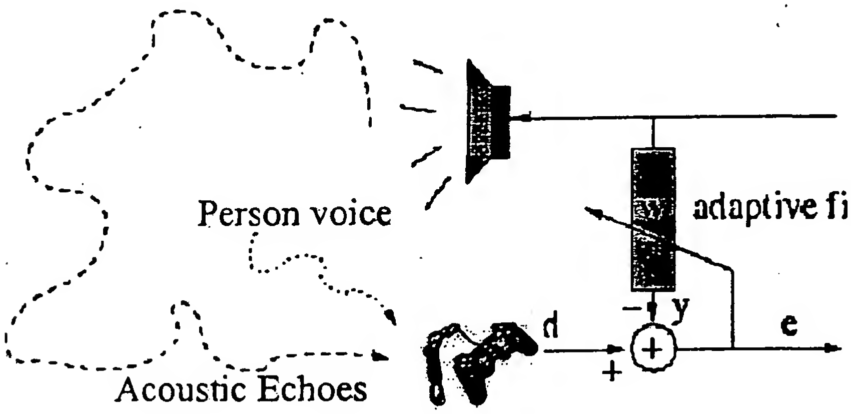


Figure3

Because one can intercept the audio signal being played on console thru either analog or digital, that intercepted signal is a noise template that can be subtracted from the microphone signal by adaptive filtering technology to produce clean signal

The AEC implemented support multi-channel sound output, such as stereo or 5.1

2. Array Beam-forming
-- To suppress signal not coming from the listening direction

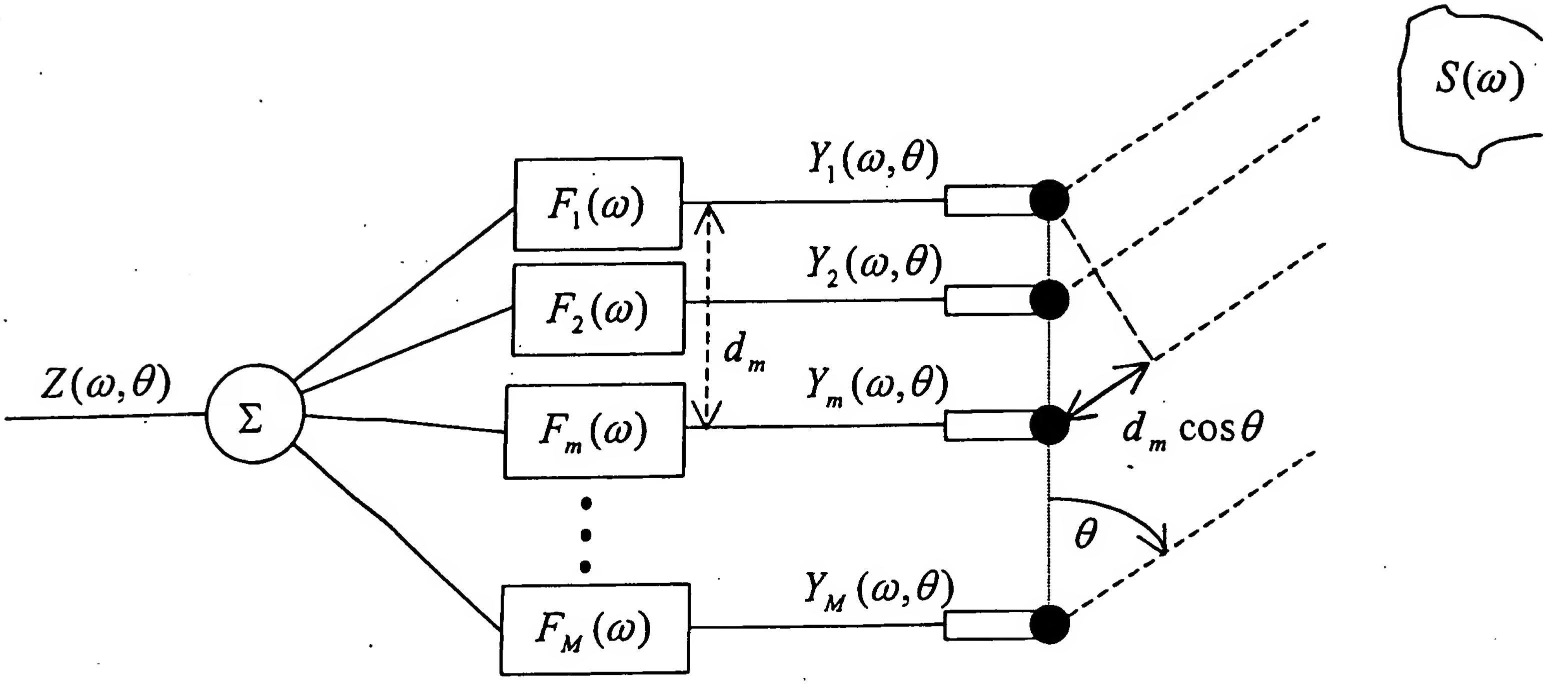


Figure4 illustrates a filter-and-sum beamformer

The Beam-forming designed is based on Filter-and-Sum; the FIR filters (called Signal-Passing-Filter) are generated by array calibration process adaptively, thus, it is essentially an Adaptive-Beamformer that can always track and steer its beam (listening direction) toward the target voice source location without physical movement of sensor array

3. Adaptive array calibration
-- To separate interferences and target voice signal

This is the core algorithm designed specifically to target game controller usage scenario and conquer the facts 1 & 2:

1. Support 6-DOF sensor array movement in 3D space, i.e. can arbitrarily change controller orientation (the sensor array's physical steering angle), can move the controller either very far away or very close toward player during talking without perceivable performance deduction
2. Enhance voice signal and preserve its quality with little or no distortion even under extreme loud gaming environment

The algorithm is based on the idea of blind source separation using second-order statistics; [redacted]

[redacted]

[redacted] 8/9

In a real acoustic environment, the signal quality might significantly deteriorate in the presence of extreme noises and room reverberation, in such case, a typical delay-and-sum or filter-and-sum beamforming would not work practically due to the fact that the signals being received from different sensors violate the fundamental assumption of "pure delay", they are in fact subject to arbitrary distortion, thus, it is crucial to have a calibration process running on backend always tracking the change of acoustic setup in every 100 milli-second (Assume the acoustic environment is stationary within this timeframe).

Two FIR filters should be adapted during calibration:

1. The Signal-Passing-Filter that is used by Filter-and-sum beamformer to enhance the target signal, the output may still have significant amount of noise
2. The Signal-Blocking-Filter that effectively blocks the target signal and generates interferences only, the interferences are later subtracted from the formal beamforming output (with noise) thru using adaptive noise cancellation technique

4. Adaptive Noise Cancellation (ANC)
-- To subtract the interferences from beamforming output

It is pretty much the same thing as AEC except the noise templates are generated by sensor array's Signal-Blocking-Filter instead of intercepting the console sound output

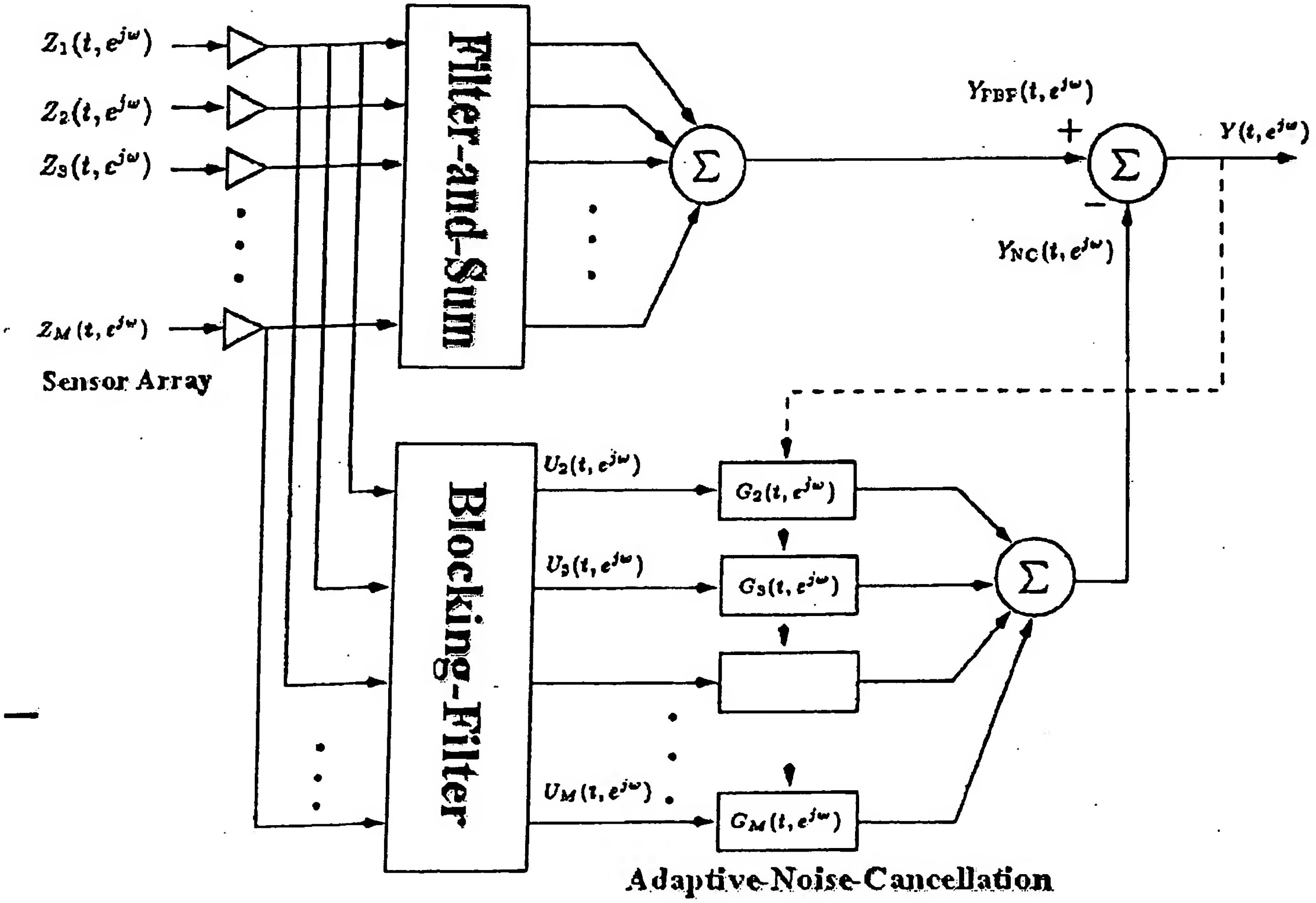
In order to maximize noise cancellation while minimizing target signal distorting, the interferences used as noise templates should prevent the target signal leakage that is covered by the Signal-Blocking-Filter

The use of ANC can attain high interference-reduction performance with a small number of microphones arranged in small space. Essentially, it conquers the fact 5: the compact mounting area likely available

[redacted]

[redacted]

Part 3: Microphone array framework



- The received microphone signal is preprocessed by AEC to remove the effect of game console sound output
- The array calibration takes place every 100 milli-second as long as the detected Signal-to-Noise-Ratio is above 0dB, it updates the Signal-Passing-Filter used in Filter-and-Sum beamformer and Signal-Blocking-Filter that generates pure interferences whose SNR is less then -100dB
- The sensor array output signal goes through post-processing to further refine the voice quality based on person-dependant voice spectrum filtering by Bayesian statistic modeling [redacted]
- The signal processing algorithms are carried out in frequency-domain; a fast and efficient FFT is the key to reach real-time signal response. The implemented software requires 25 FFT operations with size of 1024 for every signal input chunk (512 signal samples in 16 kHz sampling rate)
- In the case of Four-Sensor array with equally spaced straight line geometry, without applying AEC and Bayesian model base voice spectrum filtering, the total computation involved is about **250M Flops**

[redacted]